ELSEVIER
Artificial Intelligence 130 (2001) 1–26
Artificial Intelligence
www.elsevier.com/locate/artint

The effect of mobility on minimaxing of game trees with random leaf values

Mark Levene [a], Trevor I. Fenner [b,*]

[a] *Department of Computer Science, University College, University of London, Gower Street, London WC1E 6BT, UK*

[b] *Department of Computer Science, Birkbeck College, University of London, Malet Street, London WC1E 7HX, UK*

Received 15 October 1996; received in revised form 13 September 2000

Abstract

Random minimaxing, introduced by Beal and Smith [ICCA J. 17 (1994) 3–9], is the process of using a random static evaluation function for scoring the leaf nodes of a full width game tree and then computing the best move using the standard minimax procedure. The experiments carried out by Beal and Smith, using random minimaxing in Chess, showed that the strength of play increases as the depth of the lookahead is increased. We investigate random minimaxing from a combinatorial point of view in an attempt to gain a better understanding of the utility of the minimax procedure and a theoretical justification for the results of Beal and Smith's experiments. The concept of *domination* is central to our theory. Intuitively, one move by white dominates another move when choosing the former move would give less choice for black when it is black's turn to move, and subsequently more choice for white when it is white's turn to move. We view domination as a measure of mobility and show that when one move dominates another then its probability of being chosen is higher.

We then investigate when the probability of a "good" move relative to the probability of a "bad" move increases with the depth of search. We show that there exist situations when increased depth of search is "beneficial" but that this is not always the case. Under the assumption that each move is either "good" or "bad", we are able to state sufficient conditions to ensure that increasing the depth of search increases the strength of play of random minimaxing. If the semantics of the game under consideration match these assumptions then it is fair to say that random minimaxing appears to follow a reasonably "intelligent" strategy. In practice domination does not always occur, so it remains an open problem to find a more general measure of mobility in the absence of domination.

* Corresponding author.
E-mail addresses: mlevene@cs.ucl.ac.uk (M. Levene), trevor@dcs.bbk.ac.uk (T.I. Fenner).

0004-3702/01/$ – see front matter
PII: S0004-3702(01)00072-8

Keywords: Game trees; Minimax; Random evaluation function; Mobility; Search depth; Domination

1. Introduction

The *minimax* procedure is a fundamental search algorithm for deciding the next move to play in two-player zero-sum perfect information games between white and black [11,16, 24]; Chess, Checkers, Othello and Go are examples of such games. In order to utilise the minimax procedure, a full-width δ-ply game tree (with $\delta \geqslant 1$) is constructed, where nodes represent game positions and arcs represent legal moves from one position to another; we assume that the root node represents the current position and it is white's turn to move next, i.e., a *white position*. The minimax procedure is equipped with a static evaluation function [6], which computes a *score* for each leaf node of the constructed δ-ply game tree.

In order to evaluate the returned minimax score of the tree, the scores of the leaf nodes are backed up to the root in such a way that white maximises over the scores of its children and black minimises over the scores of its children. As we have described it, the minimax strategy is a heuristic, since a δ-ply game tree usually only contains a small fragment of the whole game, and thus the score returned by the minimax procedure is only an estimate of the true game-theoretic score of the root position. The underlying assumption of the minimax procedure is that both white and black optimise their choice of move.

Obviously, if the score of a position returned by the static evaluation function is *not* a good estimate of the strength of the position then the minimax procedure will not, in general, choose good moves. On the other hand, if the static evaluation function returns the true score of a position then only the first ply of the game tree need be examined. In practice the static evaluation function is a heuristic, and thus there ought to be a correlation between the quality of the evaluation function and the quality of the score returned by the minimax procedure. Evidence supporting this conjecture was exhibited in [4], where the minimax procedure was compared with an alternative search procedure called the *product rule*. It was shown that the minimax procedure tended to perform better than the product rule when the probability of the static evaluation function returning an erroneous score was small and worse than the product rule otherwise.

In order to measure the utility of the minimax procedure, we use a random static evaluation function that returns a natural number uniformly distributed between 1 and α inclusive [7]. This variation of the minimax procedure, called *random minimaxing*, was introduced in [3]. In this way we can decouple the effectiveness of the minimax procedure from the accuracy of the static evaluation function. The experiments carried out by Beal and Smith [3], using random minimaxing in Chess, produced the interesting result that the strength of play increases as the depth of the lookahead is increased. Herein we investigate random minimaxing from a combinatorial point of view in an attempt to gain a better understanding of the utility of the minimax procedure and a theoretical justification for the results of Beal and Smith's experiments.

A preliminary analysis of random minimaxing was carried out in [15]. Therein we showed that when $\delta = 1$ the probabilities of choosing each of the moves are the same, but when $\delta > 1$ then, in general, these probabilities are different; i.e., when $\delta > 1$ random minimaxing does *not* correspond to sampling from a uniform distribution. In [15] it was shown that, for 2-ply game trees, moves which reach nodes representing positions which have fewer children are more likely to be chosen. As a corollary of this result, we showed that in Chess (and other combinatorial games that satisfy the following assumption) random minimaxing with respect to 2-ply game trees is "stronger" than random minimaxing with respect to 1-ply game trees, under the assumption that the more a move made by white restricts black's choice of moves (i.e., black's mobility) the better that move is. We also suggested that, when $\delta > 2$, the above assumption should be extended so that, in addition, the less a move restricts white's subsequent choice of moves the better that move is; we call this extended assumption the *mobility assumption*. We observe that the mobility assumption is reasonable for many combinatorial games besides Chess such as, for example, Othello—for which restricting the opponent's choice of moves and giving oneself more choice is a well-known middle game strategy [14].

In this paper we further investigate random minimaxing for any depth of lookahead $\delta \geqslant 1$. We make the simplifying assumption that, when considering a move, each of the subgame trees rooted at nodes representing the result of choosing a move are approximated by *level-regular* subgame trees. A subgame tree is level-regular if all nodes at the same level have the same number of children. (For example, in practice Go is approximately level-regular.)

The concept of *domination* plays a central role in the theory of random minimaxing. If n is the node representing the position of the game after a possible move is made by white, then $prob(n)$ is defined to be the probability that random minimaxing on a δ-ply game tree will choose this move. Suppose we are given two nodes n_1 and n_2 representing the positions resulting from a choice of two possible moves. Informally, we say that n_1 dominates n_2 if, by white moving to n_1, in all subsequent moves black will have no more moves to choose from than if white had originally moved to n_2; similarly, in all subsequent moves white will have at least as many moves to choose from as he would have had if he had originally moved to n_2. We show that if n_1 dominates n_2 then $prob(n_1) \geqslant prob(n_2)$. That is, domination is a sufficient condition for the probability of one node to be greater than that of another; furthermore, $prob(n_1) > prob(n_2)$ if domination is *strict* (see Section 6). We call this result the *domination theorem*.

Thus, if the mobility assumption holds for the game under consideration, the domination theorem explains why random minimaxing is likely to choose "good" moves. Unfortunately, domination alone is not sufficient for deeper search to be "beneficial" (i.e., to increase the strength of play). Firstly, although domination is a sufficient condition for $prob(n_1) \geqslant prob(n_2)$ to hold, it is not a necessary condition: finding necessary and sufficient conditions remains an open problem. Furthermore, although n_1 may dominate n_2 in a δ-ply game tree, n_1 may not dominate n_2 after increasing the depth of search to a δ'-ply game tree, for some $\delta' > \delta$. We sidestep this problem by assuming that the set of moves can be partitioned into "good" moves, which lead to advantageous game positions, and "bad" moves, which lead to game positions which are not advantageous. If, for the game under consideration, random minimaxing can discriminate between "good" and "bad" moves, then it is reasonable to make the assumption that, for large enough δ, the probability of a "good" move is above average and the probability of a "bad" move is below average. Thus the first part of the *strong mobility assumption* states that every move is either "good" or "bad".

Secondly, even if n_1 is a "good" move and n_2 is a "bad" move, it may be the case that with a deeper search the probability of n_1 may actually decrease relative to that of n_2. In this case, due to the *horizon effect* [10], it may appear that increased lookahead is not "beneficial". However, further increasing the depth of lookahead would reveal that $prob(n_1)$ subsequently increases relative to $prob(n_2)$. Accordingly, the initial decrease in the ratio of the probabilities may have been due to a limited horizon, i.e., the ratio of the probabilities may not be changing monotonically as δ increases. We circumvent this problem by adding a second part to the strong mobility assumption. This states that, at deeper levels of the game tree, white's subsequent number of choices for "good" moves relative to white's subsequent number of choices for "bad" moves is above some threshold value and, correspondingly, black's subsequent number of choices for "bad" moves relative to black's subsequent number of choices for "good" moves is above some other threshold value. (Recall that we have assumed that the move is chosen by white, so whether a move is "good" or "bad" is from white's point of view.) It is then possible to show that the probability of n_1 relative to that of n_2 will increase with the depth of search, provided that the depth of lookahead is increased by at least two ply and that α is large enough.

In this case, when the strong mobility assumption holds for the game under consideration, increased lookahead is "beneficial", so random minimaxing is an effective strategy. We observe that increased lookahead seems to be beneficial in practice for many combinatorial games, such as Chess, Checkers, Othello and Go (see [22]). Although, for most combinatorial games, these assumptions will not be satisfied for all subgame trees, for many games they seem to be a reasonable approximation. (Recall that in our model we have assumed that evaluation functions are random, whereas in practice they are very sophisticated.) The domination theorem identifies a structural property of minimaxing; but, in order to take advantage of this for some particular game, we have to make appropriate semantic assumptions such as those above.

In Section 2, we review related work on the benefits of minimaxing. In Section 3, we introduce the minimax procedure and random minimaxing. In Section 4, we give the enumeration equations which are needed in order to obtain the results in the remaining sections. In Section 5, we define the probability that a given move be chosen as a result of random minimaxing, and we investigate how the probabilities of the positions resulting from choosing different moves are related. In Section 6, we define the concept of domination and prove the domination theorem, the main result of the paper. In Section 7, we investigate the effect of increased lookahead and present sufficient conditions for the probability of "good" moves to increase relative to the probability of "bad" moves. From these we obtain sufficient conditions for deeper search to be "beneficial". In Section 8, we conclude by discussing the practical relevance of our results. Finally, in Appendix A, we derive some monotonicity properties of certain functions related to the propagation function which is induced by random minimaxing; these results are used in Section 6.

2. Related work

Nau [18] investigated the probability of the last player winning when leaf nodes take one of the two values, *win* or *loss*. In his game tree model Nau assumes that the branching factor of a δ-ply game tree is uniform, say b, and that leaf node values are independent and identically distributed. In [18] it is shown that, under this model, if w_b is the unique solution in [0, 1] of the equation

$$(1-x)^b = x$$

and the probability of a leaf node being a win is greater than w_b, then the probability of the last player having a forced win tends to one as δ tends to infinity. Since w_b decreases strictly monotonically with b ($w_2 < 0.382$), in most cases the last player appears to be winning.

Our goal is to evaluate the probability of a move, which is defined as the proportion of times this move is on the principal variation when backing up is done according to the minimax rule. Therefore, assuming the evaluation function is random and the players are using the minimax rule, the probability of a move is the expected proportion of times this move will be chosen in actual play. We call such a scenario *random minimaxing*. Thus in our model a uniform branching factor is uninteresting since in this case it is evident that all moves have the same probability. In this sense our model generalises Nau's model to game trees where the branching factors of different moves may be different. This corresponds more closely to the situation in real games. Ultimately we would like to determine when a move of higher probability corresponds to a "better" move. This would provide a theoretical justification for the results of Beal and Smith's experiments [3], which show that a player using random minimaxing is stronger than a player choosing random moves uniformly from the available selection of moves. Our results indicate that, under the mobility assumption stated in the introduction, random minimaxing corresponds to increasing the first player's mobility whilst restricting the second player's mobility.

In [19,21] (cf. [1]), it was shown that the behaviour of the minimax procedure is almost always "pathological" for the uniform game tree model when the errors made by the evaluation function are independent, identically distributed and above a certain low threshold. The term *pathological* means that, as the depth of lookahead increases, the probability that a move chosen by the minimax procedure is correct tends to the probability that a randomly chosen move is correct. We conjecture that one of the reasons for the observed pathology is the assumption that the game trees are uniform (for a discussion on some of the causes of pathology for uniform game trees see the analyses in [12,19,21]). The results in [17] support this conjecture; it was shown there that, under similar assumptions to those made in [19,21], nonuniform game trees are not pathological when the number of children of a node is geometrically distributed.

Schrüfer [23] uses a model similar to that of Nau, i.e., game trees have a uniform branching factor b. However, out of the b possible moves, the number of winning moves m, instead of being constant, is taken to be a random variable M. In addition, the errors attached to the heuristic values returned by the evaluation function are modelled by two different probabilities: the probability of assigning a leaf value a win when it is a loss and the probability of assigning a leaf value a loss when it is a win. (In [19] these two types of error have the same distribution.) Schrüfer is interested in determining when the errors decrease with increased depth of search; in this case the game tree is called *deepening-positive*. The main result in [23] is that the game tree is deepening-positive if the probability that $M = 1$ is less than $1/b$. That is, the errors will decrease with deeper search if the probability of having only a single "good" candidate move is small enough.

The focus of our interest is different from that of Schrüfer. As stated above, the uniform branching factor model is uninteresting in our case when considering the probability of choosing between different moves. Moreover, we consider a range of possible values between 1 and α rather than just two values. We are not investigating the reliability of the minimax evaluation as does Schrüfer, but rather how the choice of move made is determined by the non-uniform structure of the game tree.

Baum and Smith [2] claim an improvement to the minimax algorithm by backing up probability distributions rather than single heuristic values (see also [20]). The backing up of these distributions is done via the product rule and the final choice of move is the one with the highest expected value. Baum and Smith concentrate on the algorithmics necessary to make their approach workable and do not investigate how properties of the game tree affect the choice of move. They also provide experimental evidence that their algorithm is often competitive with standard minimax implementations. In this sense their work is orthogonal to ours, since we are interested in understanding why minimax works and do not address the algorithmic issues.

Our work is closely related to the work of Hartmann [9], who attempted to understand the notion of mobility in Chess and its correlation with the probability of winning. (Hartmann's approach builds on the seminal work of Slater [25] and de Groot [5].) Hartmann's comprehensive analysis essentially showed a strong correlation between a player's number of degrees of freedom, i.e., choices per move, and the proportion of games they had won. Our results are compatible with Hartmann's since we show that minimax favours moves which lead to a consistent advantage in terms of the degrees of freedom a player has in subsequent moves. Hartmann's conclusions, as well as those of his predecessors, were based on summaries of individual moves and did not directly test how strong the correlation is between winning and maintaining a high mobility for several consecutive moves. We can only conjecture that this correlation is strong; the domination theorem essentially implies that this is a sufficient condition for maintaining an advantage.

3. Random minimaxing

We assume that the reader is familiar with the basic minimax procedure [11]. However, we recall some of the definitions given in [15] which will be relevant to this paper. A game tree T is a special kind of tree, whose nodes represent game positions and arcs represent legal moves from one position to another; the *root* node represents the current position. In general, we will not distinguish between the nodes and the positions they represent nor between the arcs and the moves they represent. Furthermore, when no confusion arises, we will refer to the position arrived at as a result of making a move as the move itself. We are assuming a two-player zero-sum perfect information game between the first player, called *white*, and the second player, called *black*, where the game has three possible outcomes:

win for white (i.e., loss for black), loss for white (i.e., win for black), or draw (see [8] for a precise definition of a game).

The *level* of a node n in T is defined recursively as follows: if n is the root node of T then its level is zero, otherwise the level of n is one plus the level of its parent node. Nodes of T which are at even levels are called *max-nodes* and nodes of T which are at odd levels are called *min-nodes*. At a max-node it is white's turn to move and at a min-node it is black's turn to move. We assume that T is a δ-ply game tree, with $\delta \geqslant 1$, where the number of ply in T is one less than the number of levels of T. Non-leaf nodes of a game tree are called *internal nodes*. A game tree satisfying the condition that each internal node has an arc for each possible legal move from the position represented by that node is called a *full-width* game tree. We will assume that all game trees are full-width game trees.

Given a game tree T and a node n in T, we define the following:

(1) $par(n)$ is the parent node of n.
(2) $ch(n)$ is the set of child nodes of n; if n is a leaf node then $ch(n) = \emptyset$. We denote the cardinality of this set, i.e., $|ch(n)|$, by $\chi(n)$.
(3) $sib(n) = ch(par(n)) - \{n\}$, i.e., the set of sibling nodes of n.
(4) *root*(T) is the single root node of T.
(5) *leaves*(T) is the set of leaf nodes of T.
(6) *moves*(T) $=$ *ch*(*root*(T)), i.e., the set of nodes representing the possible positions arrived at after white makes a single move.
(7) T$[n]$ is the subgame tree of T rooted at a node n in T; if $n =$ *root*(T), then T$[n] =$ T. The number of leaf nodes of T$[n]$, i.e., $|leaves(\mathrm{T}[n])|$, is denoted by #T$[n]$; for convenience, #T$[par(n)]$ will be denoted by $N(n)$; thus, if $n \in moves(\mathrm{T})$, $N(n) = \#\mathrm{T}$.

We let minimax(T, $\delta, score, \alpha$) denote a procedure which returns the leaf node of the *principal variation* [11] chosen by minimaxing, where T is the δ-ply game tree whose root represents the current position, *score* is a *static evaluation function*, and α is a natural number representing the maximum possible score. The principal variation is the path from the root to the leaf node returned by minimax(T, $\delta, score, \alpha$). We assume that the scoring of leaf nodes is computed by the function *score*, which returns a natural number between 1 and α inclusive. For the purpose of scoring we assume that all leaf nodes are distinct, although in practice two distinct leaf nodes may represent the same position (for example, through a transposition of moves [16]).

In general, it is possible that there is more than one principal variation, in which case the minimax procedure returns the set of leaf nodes of all the principal variations. This does not cause us any problems, since we will only be interested in knowing whether a particular leaf node, say n_0, is returned by the minimax procedure or not.

The score assigned to an internal node n of T during the evaluation of minimax(T, $\delta, score, \alpha$) is called the *backed up* score of n and is denoted by $sc(n)$; so when n is a leaf node $sc(n) = score(n)$. The backed up score of a subgame tree T$[n]$ is $sc(n)$, the score of its root n.

For random minimaxing we assume the availability of a probabilistic function *random*(α) that returns a natural number uniformly distributed between 1 and α inclusive; *random*(α) corresponds to rolling an unbiased α-sided dice. We will use the function *random*(α) as a static evaluation function for scoring the leaf nodes of a game tree.

For given $\delta, score$ and α, the procedure minimax(T, $\delta, score, \alpha$) defines a strategy for playing a particular combinatorial game. We call such a strategy the *game playing automaton* defined by minimax(T, $\delta, score, \alpha$). We will denote the (stochastic) game playing automaton defined by minimax(T, $\delta, random, \alpha$) by $\mathcal{A}^{(\delta,\alpha)}$ (or simply $\mathcal{A}^{\delta}$ when α is understood from context), where T is determined by the current position and δ, and where α is fixed. We will refer to the evaluation of minimax(T, $\delta, random, \alpha$) by $\mathcal{A}^{\delta}$ as *random minimaxing*.

We are interested in investigating the probability that $\mathcal{A}^{\delta_1}$*wins* against $\mathcal{A}^{\delta_2}$, i.e., that $\mathcal{A}^{\delta_1}$ is a stronger player than $\mathcal{A}^{\delta_2}$, where $\delta_1 \geqslant \delta_2$, under the assumption that it is decided randomly which player will start. We denote this probability by $win(\delta_1, \delta_2)$, where here we discount drawn games. Obviously, $win(\delta, \delta) = 1/2$, so we will assume from now on that $\delta_1 > \delta_2$.

We will also assume that leaf nodes which represent *terminal* positions, i.e., positions which are won, lost or drawn for the player to move, are also evaluated randomly. We make the following modification to a δ-ply game tree T which has a leaf node representing a terminal position at a level $\delta_0 < \delta$. We extend T by adding nodes representing *dummy positions* and arcs representing *dummy moves* in such a way that all the leaf nodes are at level δ; T is extended in such a way that every internal node representing a terminal or dummy position has exactly one child. We call such an extension a *canonical δ-level extension* of T and assume from now on that all δ-ply game trees are canonically δ-level extended.

The motivation for evaluating terminal nodes randomly is to avoid giving an unfair advantage to $\mathcal{A}^{\delta_1}$ simply because a terminal position has been reached and the outcome of the game may be recognized. Our approach to evaluating terminal positions is different from that taken in [3]; there, terminal positions are recognized and evaluated as a win, loss or draw according to the rules of the game. In order not to give an unfair advantage to $\mathcal{A}^{\delta_1}$, they augment $\mathcal{A}^{\delta_2}$ with the additional capability of recognizing terminal positions up to level δ_1.

We recall from the introduction that a subgame tree is *level-regular* if all nodes at the same level have the same number of children. At any stage, when considering the possible moves to be chosen from *moves*(T), we make the simplifying assumption that for all nodes $n \in$ *moves*(T) the subgame trees T[n] are level-regular. This *level-regularity assumption* makes the ensuing analysis more tractable. It is more general than assuming that game trees are uniform (i.e., that the number of children of each node is constant) [11], and also more realistic since it distinguishes between the number of choices for the two players. It is our view that level-regularity is a reasonable approximation for game trees: level-regular trees can be viewed as the result of averaging out the number of children per node for internal nodes on a given level.

4. Enumeration equations

We now give equations for enumerations which are needed in order to obtain the results in the remaining sections. We assume from now on that $n \neq root$(T) is a node in δ-ply game tree T; we often write $\hat{n}$ to indicate that n is a max-node or $\bar{n}$ to indicate that n is a min-node. For a given node n which is not a leaf of T, we assume that $n' \in ch(n)$, and we also let $m = \chi(\hat{n})$ or $q = \chi(\bar{n})$, as appropriate. We also assume that i is a natural number between 1 and α inclusive; i denotes a possible score of any of the leaf nodes returned by the minimax procedure.

We let $EQ(n,i)$ be the total number of possible subgame trees T[n] such that $sc(n) = i$, i.e., the number of assignments of scores to the leaf nodes of T[n] such that $sc(n) = i$. We shall also use counting functions $LE(n,i), LT(n,i), GE(n,i)$ and $GT(n,i)$ with the obvious meanings.

We define $LE(n,i)$ and $LT(n,i)$ as follows:

$$LE(n,i) = \sum_{j=1}^{i} EQ(n, j), \qquad LT(n,i) = LE(n, i-1).$$

Thus,

$$EQ(n,i) = LE(n,i) - LE(n,i-1) = LT(n,i+1) - LT(n,i). \tag{1}$$

The next lemma follows from (1) and the fact that $LE(n,0) = 0$.

Lemma 4.1. *$EQ(n,i) = LE(n,i) = LT(n,i+1)$ if and only if $i = 1$.*

We further define $GE(n,i)$ and $GT(n,i)$ as follows:

$$GE(n,i) = \sum_{j=i}^{\alpha} EQ(n, j), \qquad GT(n,i) = GE(n, i+1).$$

Thus,

$$EQ(n,i) = GE(n,i) - GE(n,i+1) = GT(n,i-1) - GT(n,i). \tag{2}$$

The next lemma follows from (2) and the fact that $GT(n,\alpha) = 0$.

Lemma 4.2. *$EQ(n,i) = GE(n,i) = GT(n,i-1)$ if and only if $i = \alpha$.*

From the semantics of min and max, we obtain the following equations when n is a non-leaf node.

$$LE(\hat{n},i) = LE(\bar{n}',i)^m, \tag{3}$$

$$GT(\bar{n},i) = GT(\hat{n}',i)^q. \tag{4}$$

The following equations are derived from Eqs. (3) and (4) on using Eqs. (1) and (2), respectively

$$EQ(\hat{n},i) = LE(\bar{n}',i)^m - LE(\bar{n}',i-1)^m,$$

$$EQ(\bar{n},i) = GT(\hat{n}',i-1)^q - GT(\hat{n}',i)^q.$$

When n is a leaf node $EQ(n,i) = 1$ and thus

$$LE(\bar{n},i) = i, \qquad GT(\hat{n},i) = \alpha - i.$$

The next lemma is immediate since $\alpha^{\#\mathrm{T}[n]}$ is the total number of possible subgame trees rooted at n.

Lemma 4.3. *For all nodes n in* T *and all i, $0 \leqslant i \leqslant \alpha$,*

$$\alpha^{\#\mathrm{T}[n]} = LE(n,i) + GT(n,i),$$

in particular,

$$\alpha^{\#\mathrm{T}[n]} = LE(n,\alpha) = GT(n,0).$$

The next two lemmas now follow from Lemma 4.3 on using Eqs. (3) and (4), respectively.

Lemma 4.4. *If n is a non-leaf min-node then, for all i, $0 \leqslant i \leqslant \alpha$,*

$$LE(\bar{n},i) = GT(\hat{n}',0)^q - GT(\hat{n}',i)^q.$$

Lemma 4.5. *If n is a non-leaf max-node then, for all i, $0 \leqslant i \leqslant \alpha$,*

$$GT(\hat{n},i) = LE(\bar{n}',\alpha)^m - LE(\bar{n}',i)^m.$$

In a similar fashion to the correspondence between Lemmas 4.4 and 4.5, we observe that any formula given for min-nodes has a corresponding *dual* in terms of max-nodes.

5. The probability of a node under random minimaxing

We formalise the definition of $prob(n)$ given in the introduction and extend it to apply to all nodes n in T. We view $prob(n)$ as the proportion of assignments of scores to the leaf nodes of $\mathrm{T}[par(n)]$ such that n is on a principal variation of $\mathrm{T}[par(n)]$. Using this definition of $prob(n)$, we obtain a sufficient condition for $prob(n_1)$ to be greater than $prob(n_2)$ for two nodes $n_1, n_2 \in moves(\mathrm{T})$. We also obtain an expression which allows us to recursively determine whether $prob(n_1) > prob(n_2)$.

Let $\#[n](i)$ denote the number of distinct assignments of scores to the leaf nodes of $\mathrm{T}[par(n)]$ such that $sc(par(n)) = sc(n) = i$, i.e., such that n is on a principal variation of $\mathrm{T}[par(n)]$ and $sc(n) = i$. Now let

$$\#[n] = \sum_{i=1}^{\alpha} \#[n](i).$$

Recalling that $N(n)$ denotes $\#\mathrm{T}[par(n)]$, so $\alpha^{N(n)}$ is the total number of assignments of scores to the leaf nodes of $\mathrm{T}[par(n)]$, we define $prob(n)$ the probability of a node n in T as follows:

$$prob(n) = \sum_{i=1}^{\alpha} \frac{\#[n](i)}{\alpha^N} = \frac{\#[n]}{\alpha^N},$$

where for simplicity we write N for $N(n)$.

Essentially, $prob(n)$ is the conditional probability that n is on a principal variation of T given that $par(n)$ is on a principal variation. We note that the sum of $prob(n^*)$ for all $n^* \in ch(par(n))$ is greater than or equal to one. However, it is important to observe that this sum may be strictly greater than one, since the events that two n^*'s are on (distinct) principal variations are *not mutually exclusive* [7]. For example, if by chance all the leaf nodes are assigned the same score, then minimax(T, δ, *random*, α) returns all of the leaf nodes.

We say that α is *large* when α is large in comparison to N^2. Now, assuming that α is large, it follows that the probability of two leaf nodes being assigned the same value is small. The probability that an assignment of scores to the leaf nodes of T[$par(n)$] assigns different values to distinct leaf nodes is given by

$$\frac{\alpha!}{\alpha^N(\alpha - N)!}.$$

This probability gets closer to 1 as α increases—this follows on using Stirling's approximation [13] when, after some manipulation, we obtain

$$\frac{\alpha!}{\alpha^N(\alpha - N)!} \approx \mathrm{e}^{-N}\left(1 - \frac{N}{\alpha}\right)^{-\alpha+N-1/2} \approx \mathrm{e}^{-N^2/2\alpha}. \tag{5}$$

We observe that, when α is large, the exponent of e in (5) is close to zero and thus the probability that all leaf nodes are assigned different values is close to one. Therefore, if α is large, the sum of $prob(n^*)$ for all $n^* \in ch(par(n))$ is close to one.

The following lemma gives a constructive way of computing $\#[n](i)$ when n is a min-node.

Lemma 5.1. *For all i, $1 \leqslant i \leqslant \alpha$,*

$$\#[\bar{n}](i) = EQ(\bar{n}, i) \prod_{\bar{n}^* \in sib(\bar{n})} LE(\bar{n}^*, i).$$

The next lemma is immediate from Lemma 5.1 if, when comparing the probabilities of two nodes $\bar{n}_1, \bar{n}_2 \in moves$(T), we divide $\#[\bar{n}_j](i)$, for $j = 1$ and 2, by

$$\prod_{\bar{n}^* \in moves(\mathrm{T})} LE(\bar{n}^*, i).$$

Lemma 5.2. *If $\bar{n}_1, \bar{n}_2 \in moves$(T) and*

$$\frac{EQ(\bar{n}_1, i)}{LE(\bar{n}_1, i)} \geqslant \frac{EQ(\bar{n}_2, i)}{LE(\bar{n}_2, i)} \tag{6}$$

for all $i \in \{1, 2, \ldots, \alpha\}$, then $prob(\bar{n}_1) \geqslant prob(\bar{n}_2)$. This inequality is strict if inequality (6) *is strict for some i.*

The converse of Lemma 5.2 is shown to be false by the following counterexample. Let T be a 3-ply game tree having two moves a and b, with $\chi(\mathsf{a}) = 2$, $\chi(\mathsf{b}) = 1$,

$\chi(\mathsf{a}') = 3$ for $\mathsf{a}' \in ch(\mathsf{a})$, and $\chi(\mathsf{b}') = 2$ for $\mathsf{b}' \in ch(\mathsf{b})$. For $\alpha = 3$, it can be verified that $prob(\mathsf{a}) < prob(\mathsf{b})$, but (6) does not hold for $i = 2$ (see Example 2 in Section 6).

We note that the definition of $prob(n)$ and Lemmas 5.1 and 5.2 do not depend on the level-regularity assumption.

For the rest of this section, let n_0 be a node in *moves*(T), let n be a node in T$[n_0]$ and let k be the level of n in T. (We recall that if n is a min-node then k is odd and if n is a max-node then k is even.) Since by assumption T$[n_0]$ is level regular, $\chi(n)$ depends only on k, for a given n_0. Thus we are able to write $E_k(i)$ for $EQ(n,i)$, $L_k(i)$ for $LE(n,i)$ and $G_k(i)$ for $GT(n,i)$. When n is *not* a leaf node we assume that $n' \in ch(n)$, and let $m_k = \chi(\hat{n})$ or $q_k = \chi(\bar{n})$.

The next lemma follows from Lemmas 4.4 and 4.5.

Lemma 5.3. *The following equations hold for all* i, $0 \leqslant i \leqslant \alpha$,

(i) *for odd* $k < \delta - 1$, $L_k(i) = L_{k+2}(\alpha)^{m_{k+1}q_k} - (L_{k+2}(\alpha)^{m_{k+1}} - L_{k+2}(i)^{m_{k+1}})^{q_k}$,

(ii) *for even* $k < \delta - 1$, $G_k(i) = G_{k+2}(0)^{q_{k+1}m_k} - (G_{k+2}(0)^{q_{k+1}} - G_{k+2}(i)^{q_{k+1}})^{m_k}$.

Suppose that k is odd. Let m and q be abbreviations for m_{k+1} and q_k, respectively, and let $l_k^i(\bar{n}) = L_k(i)/L_k(\alpha)$; we abbreviate $l_k^i(\bar{n})$ to l_k^i whenever $\bar{n}$ is evident from context. Then l_k^i is strictly increasing in i for $0 \leqslant i \leqslant \alpha$, since $L_k(i)$ is strictly increasing. Using Lemma 5.3(i), we obtain the following recurrence for l_k^i:

$$\begin{aligned} l_k^i &= 1 - \frac{(L_{k+2}(\alpha)^m - L_{k+2}(i)^m)^q}{L_{k+2}(\alpha)^{mq}} \\ &= 1 - \bigl(1 - \bigl(l_{k+2}^i\bigr)^m\bigr)^q. \end{aligned} \tag{7}$$

Suppose now that k is even. Now let m and q be abbreviations for m_k and q_{k+1}, respectively, and let $g_k^i(\hat{n}) = G_k(i)/G_k(0)$; we abbreviate $g_k^i(\hat{n})$ to g_k^i whenever $\hat{n}$ is evident from context. Then g_k^i is strictly decreasing in i for $0 \leqslant i \leqslant \alpha$, since $G_k(i)$ is strictly decreasing. Using Lemma 5.3(ii), we obtain a corresponding recurrence for g_k^i:

$$g_k^i = 1 - \bigl(1 - \bigl(g_{k+2}^i\bigr)^q\bigr)^m. \tag{8}$$

We can combine l_k^i and g_k^i into a single function f_k^i (which is an abbreviation of $f_k^i(n)$) defined as follows:

$$f_k^i = \begin{cases} l_k^i & \text{if } k \text{ is odd,} \\ g_k^i & \text{if } k \text{ is even.} \end{cases}$$

Similarly, using Lemmas 4.4 and 4.5, we can combine (7) and (8) yielding

$$f_k^i = 1 - \bigl(f_{k+1}^i\bigr)^{t_k}, \quad \text{where } t_k = \begin{cases} q_k & \text{if } k \text{ is odd,} \\ m_k & \text{if } k \text{ is even.} \end{cases} \tag{9}$$

For leaf nodes, the extreme case for f_k^i is given by

$$f_\delta^i = \begin{cases} i/\alpha & \text{if } \delta \text{ is odd,} \\ (\alpha - i)/\alpha & \text{if } \delta \text{ is even.} \end{cases} \tag{10}$$

If we now define

$$\mathcal{F}_k(z) = 1 - z^{t_k}, \tag{11}$$

for $0 \leqslant z \leqslant 1$, then $f_k^i = \mathcal{F}_k(f_{k+1}^i)$. We call $\mathcal{F}_k$ the *propagation* function and observe that $\mathcal{F}_k$ is strictly decreasing in z.

Now, let

$$\mathcal{F}_\phi^*(z) = \mathcal{F}_1\big(\mathcal{F}_2\big(\cdots(\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(z)))\cdots\big)\big), \tag{12}$$

where $1 < \phi \leqslant \delta$. In the special case $\phi = 1$, we let $\mathcal{F}_1^*(z) = z$. We call $\mathcal{F}_\phi^*$ the *iterated propagation* function. It follows that

$$l_1^i = f_1^i = \mathcal{F}_\phi^*\big(f_\phi^i\big) \quad \text{for all } \phi,\ 1 \leqslant \phi \leqslant \delta. \tag{13}$$

We see from (12) and (13) that the functions $\mathcal{F}_j$ propagate the values f_j^i up the game tree T from the leaf nodes to their ancestor node in *moves*(T).

We observe that $\mathcal{F}_\phi^*$ changes *parity* after each iteration: after each application of $\mathcal{F}_j$ to the corresponding intermediate result $\mathcal{F}_{j+1}(\cdots(\mathcal{F}_{\phi-1}(z))\cdots)$, the current result $\mathcal{F}_j(\mathcal{F}_{j+1}(\cdots(\mathcal{F}_{\phi-1}(z))\cdots))$, will be strictly increasing or decreasing in z according to whether the intermediate result is strictly decreasing or increasing, respectively. Thus, if ϕ is even $\mathcal{F}_\phi^*$ is strictly decreasing, but if ϕ is odd $\mathcal{F}_\phi^*$ is strictly increasing.

Let $\bar{n} \in$ *moves*(T). Then, by using (1) and (13), we can rewrite the expression $EQ(\bar{n}, i)/LE(\bar{n}, i)$ appearing in Lemma 5.2 as follows:

$$\frac{EQ(\bar{n}, i)}{LE(\bar{n}, i)} = \frac{E_1(i)}{L_1(i)} = \frac{\frac{L_1(i)}{L_1(\alpha)} - \frac{L_1(i-1)}{L_1(\alpha)}}{\frac{L_1(i)}{L_1(\alpha)}} = 1 - \frac{l_1^{i-1}}{l_1^i} = 1 - \frac{\mathcal{F}_\phi^*(f_\phi^{i-1})}{\mathcal{F}_\phi^*(f_\phi^i)}, \tag{14}$$

for any level $\phi,\ 1 \leqslant \phi \leqslant \delta$.

6. Domination

The concept of *domination* plays an important role in the theory of random minimaxing. Let $n_1, n_2 \in$ *moves*(T) and let r_1 and r_2 denote nodes at the same level in T[n_1] and T[n_2], respectively. Then n_1 *dominates* n_2, written $n_1 \succeq n_2$, if, for all such r_1 and r_2, $\chi(r_1) \leqslant \chi(r_2)$ when r_1 and r_2 are min-nodes and $\chi(r_1) \geqslant \chi(r_2)$ when r_1 and r_2 are max-nodes. Moreover, n_1 *strictly dominates* n_2, written $n_1 \succ n_2$, if $n_1 \succeq n_2$ and $\chi(r_1) \neq \chi(r_2)$ for some r_1 and r_2. Finally, n_1 *dominates* n_2 *solely* at level ϕ, written $n_1 \succ_\phi n_2$, if $n_1 \succ n_2$ and $\chi(r_1) \neq \chi(r_2)$ for all r_1 and r_2 at level ϕ, but $\chi(r_1^*) = \chi(r_2^*)$ for all nodes r_1^* in T[n_1] and r_2^* in T[n_2] at any other level.

The main result of this section utilises the monotonicity results proved in Appendix A to show that if n_1 dominates n_2 then *prob*(n_1) $\geqslant$ *prob*(n_2); if domination is *strict* then this inequality is strict. Thus domination is a sufficient condition for the probability of one move to be greater than or equal to that of another.

Lemma 6.1. *For any nodes n_1 and n_2 in moves(T), if $n_1 \succ_\phi n_2$ for some $\phi,\ 1 \leqslant \phi < \delta$, then prob($n_1$) $>$ prob(n_2).*

Proof. By Lemma 5.2 it is sufficient to show that

$$\frac{EQ(\bar{n}_1, i)}{LE(\bar{n}_1, i)} > \frac{EQ(\bar{n}_2, i)}{LE(\bar{n}_2, i)},$$

for $1 < i \leqslant \alpha$. Let r_1 and r_2 be nodes at level ϕ in T[n_1] and T[n_2], respectively. Now n_1 and n_2 are moves and are thus at level 1. So, by (14), the above inequality is equivalent to

$$1 - \frac{\mathcal{F}^*_\phi(f^{i-1}_\phi(r_1))}{\mathcal{F}^*_\phi(f^{i}_\phi(r_1))} > 1 - \frac{\mathcal{F}^*_\phi(f^{i-1}_\phi(r_2))}{\mathcal{F}^*_\phi(f^{i}_\phi(r_2))}. \tag{15}$$

In effect, we are considering the backed up values of $f^i_\phi(r_j)$ and $f^{i-1}_\phi(r_j)$, for $j = 1$ and 2, to be the values of the corresponding leaf nodes of a ϕ-ply game tree.

Now $\phi < \delta$, so let x denote $f^{i-1}_{\phi+1}(r'_j)$ and y denote $f^{i}_{\phi+1}(r'_j)$, where $r'_j \in ch(r_j)$. (Note that these values are the same for $j = 1$ and $j = 2$, since n_1 dominates n_2 solely at level ϕ.)

Assume that ϕ is odd; then r_1 and r_2 are min-nodes, so r'_1 and r'_2 are max-nodes. It then follows that $0 < y < x < 1$, since $g^i_{\phi+1}$ is strictly decreasing in i. On using Eqs. (9) and (11), we obtain

$$f^i_\phi(\bar{r}_j) = l^i_\phi(\bar{r}_j) = \mathcal{F}_\phi\big(g^i_{\phi+1}(\hat{r}'_j)\big) = 1 - \big(g^i_{\phi+1}(\hat{r}'_j)\big)^{q_{\phi j}} = 1 - y^{q_{\phi j}}, \tag{16}$$

where $q_{\phi j} = \chi(\bar{r}_j)$, for $j = 1$ and 2. Similarly,

$$f^{i-1}_\phi(\bar{r}_j) = l^{i-1}_\phi(\bar{r}_j) = \mathcal{F}_\phi\big(g^{i-1}_{\phi+1}(\hat{r}'_j)\big) = 1 - \big(g^{i-1}_{\phi+1}(\hat{r}'_j)\big)^{q_{\phi j}} = 1 - x^{q_{\phi j}}. \tag{17}$$

Therefore,

$$\frac{f^{i-1}_\phi(\bar{r}_j)}{f^{i}_\phi(\bar{r}_j)} = \frac{\mathcal{F}_\phi(g^{i-1}_{\phi+1}(\hat{r}'_j))}{\mathcal{F}_\phi(g^{i}_{\phi+1}(\hat{r}'_j))} = \frac{1 - x^{q_{\phi j}}}{1 - y^{q_{\phi j}}}.$$

Now $0 < q_{\phi 1} < q_{\phi 2}$ since $n_1 \succ_\phi n_2$. So, on using part (ii) of Corollary A.7, we see that

$$\frac{1 - x^{q_{\phi 1}}}{1 - y^{q_{\phi 1}}} < \frac{1 - x^{q_{\phi 2}}}{1 - y^{q_{\phi 2}}}, \tag{18}$$

or equivalently

$$\frac{f^{i-1}_\phi(\bar{r}_1)}{f^{i}_\phi(\bar{r}_1)} < \frac{f^{i-1}_\phi(\bar{r}_2)}{f^{i}_\phi(\bar{r}_2)}. \tag{19}$$

Letting $x_1 = 1 - x^{q_{\phi 1}}$, $x_2 = 1 - x^{q_{\phi 2}}$, $y_1 = 1 - y^{q_{\phi 1}}$ and $y_2 = 1 - y^{q_{\phi 2}}$, it follows that $x_2 < y_2$ and $y_1 < y_2$. Also, inequality (18) now becomes

$$\frac{x_1}{y_1} < \frac{x_2}{y_2}. \tag{20}$$

We now utilise Corollary A.4 to obtain, for any $m, q \geqslant 1$,

$$\frac{1 - (1 - x_1^m)^q}{1 - (1 - y_1^m)^q} < \frac{1 - (1 - x_2^m)^q}{1 - (1 - y_2^m)^q}.$$

By invoking a two step propagation of $\mathcal{F}$, we have

$$\frac{\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(x_j))}{\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(y_j))} = \frac{1-(1-x_j^{m_{\phi-1}})^{q_{\phi-2}}}{1-(1-y_j^{m_{\phi-1}})^{q_{\phi-2}}},$$

and thus

$$\frac{\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(x_1))}{\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(y_1))} < \frac{\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(x_2))}{\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(y_2))}. \tag{21}$$

Moreover, by Eq. (12) we have

$$\mathcal{F}_\phi^*\big(f_\phi^i(r_j)\big) = \mathcal{F}_1\big(\mathcal{F}_2\big(\cdots(\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(f_\phi^i(r_j))))\cdots\big)\big),$$

and similarly for $\mathcal{F}_\phi^*(f_\phi^{i-1}(r_j))$. We now repeatedly invoke a two step propagation of $\mathcal{F}$ and use Corollary A.4, as in the derivation of (21) from (20). Remembering that ϕ is odd, this therefore yields

$$\begin{aligned}&\frac{\mathcal{F}_1(\mathcal{F}_2(\cdots(\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(f_\phi^{i-1}(r_1))))\cdots))}{\mathcal{F}_1(\mathcal{F}_2(\cdots(\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(f_\phi^{i}(r_1))))\cdots))}\\ &\quad< \frac{\mathcal{F}_1(\mathcal{F}_2(\cdots(\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(f_\phi^{i-1}(r_2))))\cdots))}{\mathcal{F}_1(\mathcal{F}_2(\cdots(\mathcal{F}_{\phi-2}(\mathcal{F}_{\phi-1}(f_\phi^{i}(r_2))))\cdots))},\end{aligned} \tag{22}$$

and thus we deduce that

$$\frac{\mathcal{F}_\phi^*(f_\phi^{i-1}(r_1))}{\mathcal{F}_\phi^*(f_\phi^{i}(r_1))} < \frac{\mathcal{F}_\phi^*(f_\phi^{i-1}(r_2))}{\mathcal{F}_\phi^*(f_\phi^{i}(r_2))},$$

yielding (15) as required.

Now assume that ϕ is even; so r_1 and r_2 are now max-nodes, and thus r_1' and r_2' are min-nodes. It then follows that $0 < x < y < 1$, since $l_{\phi+1}^i$ is strictly increasing in i.

Corresponding to (16) and (17) we now obtain

$$f_\phi^i(\hat{r}_j) = g_\phi^i(\hat{r}_j) = \mathcal{F}_\phi\big(l_{\phi+1}^i(\bar{r}_j')\big) = 1-\big(l_{\phi+1}^i(\bar{r}_j')\big)^{m_{\phi j}} = 1-y^{m_{\phi j}},$$

where $m_{\phi j} = \chi(\hat{r}_j)$, for $j = 1$ and 2. Similarly,

$$f_\phi^{i-1}(\hat{r}_j) = g_\phi^{i-1}(\hat{r}_j) = \mathcal{F}_\phi\big(l_{\phi+1}^{i-1}(\bar{r}_j')\big) = 1-\big(l_{\phi+1}^{i-1}(\bar{r}_j')\big)^{m_{\phi j}} = 1-x^{m_{\phi j}}.$$

We note that $m_{\phi 1} > m_{\phi 2} > 0$, since $n_1 \succ_\phi n_2$. So, on using Corollary A.5, we see that

$$\frac{1-(1-x^{m_{\phi 1}})^{q_{\phi-1}}}{1-(1-y^{m_{\phi 1}})^{q_{\phi-1}}} < \frac{1-(1-x^{m_{\phi 2}})^{q_{\phi-1}}}{1-(1-y^{m_{\phi 2}})^{q_{\phi-1}}},$$

or equivalently

$$\frac{\mathcal{F}_{\phi-1}(f_\phi^{i-1}(\hat{r}_1))}{\mathcal{F}_{\phi-1}(f_\phi^{i}(\hat{r}_1))} < \frac{\mathcal{F}_{\phi-1}(f_\phi^{i-1}(\hat{r}_2))}{\mathcal{F}_{\phi-1}(f_\phi^{i}(\hat{r}_2))},$$

corresponding to (19). The proof is now concluded as before by repeatedly invoking a two step propagation of $\mathcal{F}$ and using Corollary A.4. Remembering that ϕ is now even, this yields (22) as previously. $\square$

Theorem 6.2 (Domination theorem). *For any nodes n_1 and n_2 in moves*(T), *if $n_1 \succ n_2$ then prob$(n_1) >$ prob(n_2).*

Proof. Let d be the number of levels in T such that at each of these levels the nodes in T$[n_1]$ have a different number of children from the corresponding nodes in T$[n_2]$. We obtain the result by induction on d.

Basis: If $d = 1$, the result follows by Lemma 6.1.

Induction: Assume that the result holds for some $d \geqslant 1$. We prove that the result holds for $d+1$ levels. Let r_1 in T$[n_1]$ and r_2 in T$[n_2]$ be typical nodes at the least level ϕ of T for which $\chi(r_1) \neq \chi(r_2)$. Let n_3 be a new move such that T$[n_3]$ is isomorphic to T$[n_2]$ except that $\chi(r_3) = \chi(r_1)$ for each node r_3 at level ϕ in T$[n_3]$. Thus $n_3 \succ_\phi n_2$. Moreover, $n_1 \succ n_3$ and the numbers of children of nodes in T$[n_1]$ and T$[n_3]$ differ at precisely d levels. Therefore, *prob*$(n_3) >$ *prob*(n_2) by Lemma 6.1 and *prob*$(n_1) >$ *prob*(n_3) by the induction hypothesis, yielding the result. □

Corollary 6.3. *For any nodes n_1 and n_2 in moves*(T), *if $n_1 \succeq n_2$ then prob$(n_1) \geqslant$ prob(n_2).*

Example 1. Consider the 3-ply game tree shown in Fig. 1, in which node a strictly dominates node b, i.e., a $\succ$ b, and suppose $\alpha = 2$. From the definition of *prob* and Lemma 5.1,

$$prob(\mathsf{a}) = \frac{EQ(\mathsf{a},1)LE(\mathsf{b},1) + EQ(\mathsf{a},2)LE(\mathsf{b},2)}{2^{12}}, \tag{23}$$

since N the number of leaves is 12.

For the subtree rooted at a, $q_1 = 2$ and $m_2 = 3$. So, from Lemma 5.3 with $k = 1$, we have

$$LE(\mathsf{a}, i) = L_1(i) = L_3(2)^6 - \left(L_3(2)^3 - L_3(i)^3\right)^2.$$

Since the nodes at level 3 are leaf nodes, $L_3(i) = i$, so $LE(\mathsf{a}, i) = 2^6 - (2^3 - i^3)^2$. Similarly, for the subtree rooted at b, $q_1 = 3$ and $m_2 = 2$, so $LE(\mathsf{b}, i) = 2^6 - (2^2 - i^2)^3$.

Substituting these values into (23), and using (1) to compute EQ, gives *prob*(a) $= 3691/4096$. A similar calculation yields *prob*(b) $= 2283/4096$. Thus *prob*(a) $>$ *prob*(b),

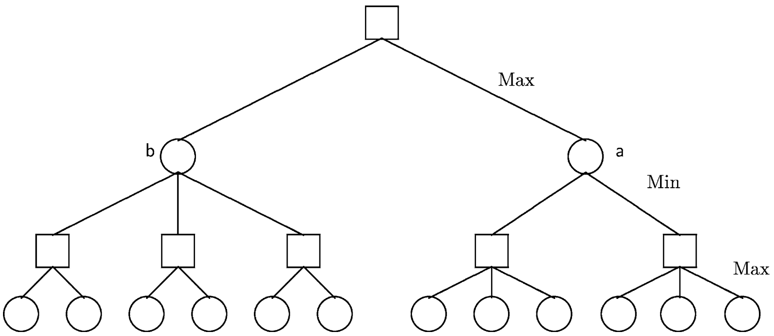

Fig. 1. A 3-ply game tree.

in accordance with the domination theorem. (In fact, by the domination theorem, $prob(\mathsf{a}) > prob(\mathsf{b})$ for any α.)

The presence of domination implies that random minimaxing will tend to choose "good" moves provided the following assumption holds.

Assumption 6.4 (*Mobility assumption*). The more a move made by white restricts black's choice of moves and the less this move restricts white's subsequent choice of moves the "better" that move is.

As we observed in the introduction, the mobility assumption is reasonable for many combinatorial games. If, for the game under consideration, the mobility assumption is valid then, when one moves dominates another, the domination theorem guarantees that the "better" move has a higher probability of being chosen.

We close this section with an example in which there is no domination.

Example 2. Let us modify the game tree shown in Fig. 1 so that, instead of $\chi(\mathsf{b}) = 3$, we have $\chi(\mathsf{b}) = 1$. So neither a nor b dominates the other. For $\alpha = 2$, it can be verified that $prob(\mathsf{a}) = 211/256$ and $prob(\mathsf{b}) = 207/256$; in this case, $prob(\mathsf{a}) > prob(\mathsf{b})$. For $\alpha = 3$, however, $prob(\mathsf{a}) = 4562/6561$ and $prob(\mathsf{b}) = 4802/6561$; in this case $prob(\mathsf{a}) < prob(\mathsf{b})$. By computation we have verified that, for $3 \leqslant \alpha \leqslant 100$, $prob(\mathsf{a})/prob(\mathsf{b})$ is monotonically decreasing in α. Moreover, it can be shown that $prob(\mathsf{a}) < prob(\mathsf{b})$ for all $\alpha > 2$.

7. Increased lookahead

Domination implies that there is a relationship between the probabilities of two moves in a single game tree, but does not take into account the effect of increased lookahead. We now investigate sufficient conditions for deeper search to be "beneficial" in the following sense. Let us assume that the set of moves can be partitioned into "good" moves which tend to lead to advantageous game positions and "bad" moves which tend to lead to game positions which are not advantageous. (Recall that a move is chosen by white, so both "good" and "bad" are from white's point of view.) If, for the game under consideration, random minimaxing can discriminate between "good" and "bad" moves, then it is reasonable to assume that the probability of "good" moves is *above average* and the probability of "bad" moves is *below average*. (The *average* probability of a move could, for example, be defined as $1/|moves(\mathrm{T})|$, but using any other reasonable formula for defining the average probability of a move does not affect the results below.)

We say that increased lookahead (i.e., deeper search) is "beneficial" if the probability of each "good" move *relative* to the probability of each "bad" move increases with the depth of search. As we shall see below, it is not always the case that increased lookahead is "beneficial" in this sense.

Let us assume that T_1 is a δ_1-ply game tree and that T_2 is a δ_2-ply game tree, where $\delta_1 > \delta_2$, such that $root(T_1)$ and $root(T_2)$ represent the same current position (i.e., T_2 consists of the first δ_2 ply of T_1). Furthermore, let $n_1^g, n_1^b \in moves(T_1)$ and $n_2^g, n_2^b \in moves(T_2)$ be such that n_1^g represents the same position as n_2^g and n_1^b represents the same position as n_2^b. We are interested in the situation when n_1^g (or equivalently n_2^g) is assumed to be any "good" move and n_1^b (or equivalently n_2^b) is assumed to be any "bad" move.

We now investigate sufficient conditions for

$$\frac{prob(n_1^g)}{prob(n_1^b)} > \frac{prob(n_2^g)}{prob(n_2^b)} \tag{24}$$

to hold, i.e., for increased lookahead to be "beneficial".

By Lemma 5.1 and the definitions of $prob$ and $l_k^i(n)$, inequality (24) is equivalent to

$$\frac{\sum_{i=1}^{\alpha} \frac{EQ(n_1^g,i)}{LE(n_1^g,i)} \prod_{n_1 \in moves(T_1)} l_1^i(n_1)}{\sum_{i=1}^{\alpha} \frac{EQ(n_1^b,i)}{LE(n_1^b,i)} \prod_{n_1 \in moves(T_1)} l_1^i(n_1)} > \frac{\sum_{i=1}^{\alpha} \frac{EQ(n_2^g,i)}{LE(n_2^g,i)} \prod_{n_2 \in moves(T_2)} l_1^i(n_2)}{\sum_{i=1}^{\alpha} \frac{EQ(n_2^b,i)}{LE(n_2^b,i)} \prod_{n_2 \in moves(T_2)} l_1^i(n_2)}. \tag{25}$$

Let r be a node at level δ_2 in T_1. Define the product of the m_k's between levels δ_2 and δ_1 in $T_1[r]$ by

$$M(r) = \prod_{k=\delta_2}^{\delta_1 - 1} m_k,$$

where we put $m_k = 1$ when k is odd. Correspondingly, define the product of the q_k's between levels δ_2 and δ_1 in $T_1[r]$ by

$$Q(r) = \prod_{k=\delta_2}^{\delta_1 - 1} q_k,$$

where we put $q_k = 1$ when k is even.

In order to ensure that both the above products are over non-trivial sets, we assume for the rest of this section that $\delta_1 \geqslant \delta_2 + 2$. Now let r_1^g be a node of $T_1[n_1^g]$ at level δ_2 and r_1^b be a node of $T_1[n_1^b]$ at level δ_2. We claim that

$$\frac{EQ(n_1^g, i)}{LE(n_1^g, i)} \nearrow 1 \quad \text{as } M(r_1^g) \to \infty \tag{26}$$

and

$$\frac{EQ(n_1^b, i)}{LE(n_1^b, i)} \searrow 0 \quad \text{as } Q(r_1^b) \to \infty, \tag{27}$$

for $1 < i \leqslant \alpha$. When $i = 1$, the left-hand sides of both (26) and (27) are equal to 1, since $EQ(n_1, 1) = LE(n_1, 1)$. We argue that we can make the left-hand side of (26) as close to one as we require and the left-hand side of (27) as close to zero as we require.

Firstly, consider (26). Assuming that $M(r_1^g)$ is large, then m_{k+1} is large for some odd k, where $\delta_2 \leqslant k + 1 < \delta_1$. Let

$$\varepsilon_1 = \left(l_{k+2}^{i-1}\right)^{m_{k+1}} \quad \text{and} \quad \varepsilon_2 = \left(l_{k+2}^{i}\right)^{m_{k+1}}.$$

We consider the case when $i = \alpha$ separately; so assume now that $i < \alpha$. Since l^i_{k+2} is strictly increasing in i and m_{k+1} is large, we have

$$0 < \varepsilon_1 \ll \varepsilon_2 \ll 1.$$

On using (9) we obtain

$$\frac{l_1^{i-1}}{l_1^i} = \frac{1-(1-(\cdots(1-(1-(1-\varepsilon_1)^{q_k})^{m_{k-1}})\cdots)^{m_2})^{q_1}}{1-(1-(\cdots(1-(1-(1-\varepsilon_2)^{q_k})^{m_{k-1}})\cdots)^{m_2})^{q_1}}.$$

Now, on using the approximation $(1-\varepsilon)^q \approx 1 - q\varepsilon$, which follows from the binomial theorem provided $q\varepsilon \ll 1$, we obtain

$$\frac{l_1^{i-1}}{l_1^i} \approx \frac{q_1(\cdots(q_k\varepsilon_1)^{m_{k-1}}\cdots)^{m_2}}{q_1(\cdots(q_k\varepsilon_2)^{m_{k-1}}\cdots)^{m_2}} = \left(\frac{\varepsilon_1}{\varepsilon_2}\right)^{m_{k-1}m_{k-3}\cdots m_2}. \tag{28}$$

Together with (14), this now yields (26), since $\varepsilon_1 \ll \varepsilon_2$. We note that the convergence is exponential in $m_2 m_4 \cdots m_{k-1} m_{k+1}$.

When $i = \alpha$ then $\varepsilon_2 = 1$ and a similar argument will confirm that (26) also holds in this case.

Secondly, consider (27). Assuming that $Q(r_1^b)$ is large, then q_{k+1} is large for some even k, where $\delta_2 \leqslant k+1 < \delta_1$. Now let

$$\varepsilon_1 = \left(g_{k+2}^{i-1}\right)^{q_{k+1}} \quad \text{and} \quad \varepsilon_2 = \left(g_{k+2}^{i}\right)^{q_{k+1}}.$$

Since g^i_{k+2} is strictly decreasing in i and q_{k+1} is large, we have

$$1 \gg \varepsilon_1 \gg \varepsilon_2 \geqslant 0.$$

Analogously to (28), we now obtain

$$\begin{aligned}\frac{l_1^{i-1}}{l_1^i} &= \frac{1-(1-(\cdots(1-(1-(1-\varepsilon_1)^{m_k})^{q_{k-1}})\cdots)^{m_2})^{q_1}}{1-(1-(\cdots(1-(1-(1-\varepsilon_2)^{m_k})^{q_{k-1}})\cdots)^{m_2})^{q_1}} \\ &\approx \frac{1-(m_2(\cdots(m_k\varepsilon_1)^{q_{k-1}})\cdots)^{q_1}}{1-(m_2(\cdots(m_k\varepsilon_2)^{q_{k-1}})\cdots)^{q_1}} = \frac{1-\mu\varepsilon_1^{q_{k-1}q_{k-3}\cdots q_1}}{1-\mu\varepsilon_2^{q_{k-1}q_{k-3}\cdots q_1}},\end{aligned} \tag{29}$$

where $\mu = m_2^{q_1} m_4^{q_3 q_1} \cdots m_k^{q_{k-1}q_{k-3}\cdots q_1}$. Together with (14), this now yields (27), and again the convergence is exponential, this time in $q_1 q_3 \cdots q_{k-1} q_{k+1}$.

Intuitively, increasing $M(r_1^g)$ increases white's mobility for "good" moves relative to white's mobility for "bad" moves. On the other hand, increasing $Q(r_1^b)$ increases black's mobility for "bad" moves relative to black's mobility for "good" moves, and thus black's mobility for "good" moves relative to black's mobility for "bad" moves decreases. If we measure the *relative mobility* of one white move compared to another white move by how much choice white has when it is white's turn to move and how little choice black has when it is black's turn to move, then increasing $M(r_1^g)$ and $Q(r_1^b)$ increases white's relative mobility for the move n_1^g compared to the move n_1^b.

Assuming that $M(r_1^g)$ and $Q(r_1^b)$ are large enough, we can approximate the left-hand side of inequality (25) by

$$\frac{\prod_{n_1 \in moves(T_1)} l_1^1(n_1) + \sum_{i=2}^{\alpha} \prod_{n_1 \in moves(T_1)} l_1^i(n_1)}{\prod_{n_1 \in moves(T_1)} l_1^1(n_1)},$$

which is greater than α since $l_1^i(n_1)$ is strictly increasing in i. (We note that the right-hand side of inequality (24), and consequently (25), is bounded as α increases; this is because, as $\alpha \to \infty$, $prob(n)$ tends to the corresponding probability of n when the scores of the leaf nodes are independent *continuous* random variables uniformly distributed on $[0, 1]$.) It follows that, provided α is sufficiently large, the left-hand side of inequality (25) will be greater than its right-hand side. Intuitively, provided α is large enough, by sufficiently increasing white's relative mobility, we can ensure that inequality (24) will hold, i.e., that the probability of a "good" move relative to the probability of a "bad" move will increase with the depth of search.

A *max-modification* of T_1 with respect to r_1^g is a game tree resulting from modifying $T_1[r_1^g]$ by increasing some of the m_k's for even k between δ_2 and $\delta_1 - 1$. Correspondingly, a *min-modification* of T_1 with respect to r_1^b is a game tree resulting from modifying $T_1[r_1^b]$ by increasing some of the q_k's for odd k between δ_2 and $\delta_1 - 1$. The above discussion is summarised in the following theorem.

Theorem 7.1 (Increased lookahead theorem). *For α sufficiently large, there exist threshold values M and Q such that, for all max-modifications of T_1 with respect to r_1^g with $M(r_1^g) \geqslant M$ and all min-modifications of T_1 with respect to r_1^b with $Q(r_1^b) \geqslant Q$, inequality* (24) *holds.*

We note that in the proof of the above theorem we only considered a restricted case of max and min modifications, where just a single m_j and q_j were increased. It is likely that a more detailed analysis would reveal that it would be sufficient to make smaller modifications at a number of levels.

It is interesting to note that, in order to prove the increased lookahead theorem, we have assumed that $\delta_1 \geqslant \delta_2 + 2$. By increasing the depth of search by at least two ply, we are able to increase white's relative mobility to a sufficient extent. This involves increasing white's relative choice and correspondingly decreasing black's relative choice. It is an open problem whether the conditions of the theorem can be relaxed, i.e., the above argument does not show whether a single threshold value, resulting from either a min-modification or a max-modification, is sufficient to ensure that increased lookahead is beneficial. A particular unresolved case of this is when $\delta_1 = \delta_2 + 1$.

We also note that it follows from (28) and (29) that the rate of convergence of the ratio of the probabilities on the left-hand side of (24) is exponential. Thus, provided α is large enough, we do not expect the threshold values implied by the increased lookahead theorem to be excessively large.

Using these results, we can now state sufficient conditions for the game playing automaton $\mathcal{A}^{\delta_1}$ to be a stronger player than $\mathcal{A}^{\delta_2}$, where $\delta_1 > \delta_2$. The result hinges upon the *strong mobility assumption* given below. If the game under consideration satisfies this assumption then, assuming that α is large enough and that $\delta_1 \geqslant \delta_2 + 2$, we can show that $win(\delta_1, \delta_2) > 1/2$, i.e., in this case random minimaxing appears to play reasonably "intelligently".

When $M(r_1^g)$ is above the threshold value M indicated in the increased lookahead theorem, we will say that white's subsequent choice for "good" moves is "much greater" than white's subsequent choice for "bad" moves". Correspondingly, when $Q(r_1^b)$ is above the threshold value Q indicated in the increased lookahead theorem, we will say that black's subsequent choice for "bad" moves is "much greater" than black's subsequent choice for "good" moves.

Assumption 7.2 (*Strong mobility assumption*). Each move is either "good" or "bad". Moreover, when increasing the lookahead, white's subsequent choice for "good" moves is "much greater" than white's subsequent choice for "bad" moves; correspondingly, black's subsequent choice for "bad" moves is "much greater" than black's subsequent choice for "good" moves.

The strong mobility assumption allows us to compare the probabilities of moves without assuming that "good" moves dominate "bad" ones. The following corollary is a direct application of the increased lookahead theorem and the above assumption.

Corollary 7.3 (Increased lookahead corollary). *Assume that α is sufficiently large and that for the game under consideration the strong mobility assumption holds. Then $win(\delta_1, \delta_2) > 1/2$ provided $\delta_1 \geqslant \delta_2 + 2$.*

Proof. Let $T_1, T_2, n_1^g, n_2^g, n_1^b$ and n_2^b be as in the previous section, where n_j^g and n_j^b range over the "good" and "bad" moves in $moves(T_j)$, respectively, for $j = 1$ and 2.

Now, since α is large, the probability that a good move will be chosen by $\mathcal{A}^{\delta_j}$ is approximately

$$P_j^g = \sum_{n_j^g} prob\big(n_j^g\big).$$

Similarly, the probability that a bad move will be chosen by $\mathcal{A}^{\delta_j}$ is approximately

$$P_j^b = \sum_{n_j^b} prob\big(n_j^b\big).$$

We need to show that $P_1^g > P_2^g$, or equivalently that $P_1^b < P_2^b$, since $P_j^g + P_j^b \approx 1$ for large α—see the discussion following (5).

If the probabilities of all "good" moves in T_1 are greater than their corresponding probabilities in T_2, then the result follows. Otherwise, suppose that $prob(n_1^g) \leqslant prob(n_2^g)$ for some n_1^g and n_2^g. As $\delta_1 \geqslant \delta_2 + 2$, the strong mobility assumption implies that the conditions stated in the increased lookahead theorem are satisfied, and thus inequality (24) holds. It follows that, for all "bad" moves n_1^b and n_2^b, we have $prob(n_1^b) < prob(n_2^b)$ and therefore $P_1^b < P_2^b$, yielding the result. $\square$

We note that if P_2^g is close to one then there is no need to increase the depth of search since $\mathcal{A}^{\delta_2}$ will almost certainly choose a "good" move. We further observe that increased lookahead seems to be beneficial in practice for many combinatorial games such as Chess, Checkers, Othello and Go (see [22]). Regarding the condition $\delta_1 \geqslant \delta_2 + 2$, it is interesting to note that, in the experiments carried out in [22], the number of ply was increased by two at each stage, since the authors claim that "it introduces more stability into the search".

8. Concluding remarks

Our analysis of random minimaxing provides some insight into the utility of the minimax procedure. Our results show that, under certain assumptions, we can closely relate the utility of the minimax procedure for game trees with random leaf values to the structure of the game tree under consideration. If the semantics of the game concerned match these assumptions, then it is fair to say that random minimaxing plays reasonably "intelligently". In particular, we have shown that if one move dominates another then its probability of being chosen is higher. Under the mobility assumption, the domination theorem (Theorem 6.2) implies that, when domination occurs in a game tree, random minimaxing is more likely to choose a "good" move. Moreover, under the strong mobility assumption, Corollary 7.3 implies that increasing the depth of search (by at least two ply) is "beneficial", provided α is large enough.

Although, in practice, we can only expect our assumptions to hold approximately, we suggest that they do provide a reasonably good model for a large class of combinatorial games. As a consequence of the domination theorem, when given the choice between two moves n_1 and n_2 where n_1 dominates n_2, a random minimaxing player will prefer n_1. This provides theoretical support to Hartmann's analysis of Chess grandmaster games [9], which showed a strong correlation between winning and having an advantage in mobility. In addition, the domination theorem gives a plausible explanation of Beal and Smith's results [3]: the reason why a random minimaxing player is stronger than a player who chooses moves according to a uniform distribution is that, in general, the former player will maximise his/her mobility.

One way of incorporating random minimaxing into game playing software is suggested by one of the experiments carried out by Beal and Smith [3]. In this experiment, one Chess program, with an evaluation function based solely on material balance, was played against another Chess program with an evaluation function based on a weighted sum of material balance and a random evaluation, where the weight of the random component was small. The results showed that including the random component in the evaluation function improved the strength of play and, moreover, the improvement increased with deeper search. Our results, considered in conjunction with these preliminary experiments, suggest that it may be beneficial to include a small random component in the evaluation functions of current game playing software.

We are currently attempting to generalise the domination theorem in order to improve our understanding of the conditions, both sufficient and necessary, for $prob(n_1)$ to be greater than $prob(n_2)$. We hope that this will allow us to obtain a more general measure of mobility than that implicit in the domination theorem. One plausible conjecture is that, for large α, the mobility of a move is related to the ratio of some product of the m_k's to a similar product of the q_k's, for $k = 1, 2, \ldots, \delta$.

Appendix A. Monotonicity properties

In this appendix we prove some fundamental results concerning the monotonicity properties of some functions closely related to the propagation function.

Lemma A.1. *For all* $0 < x \leqslant 1$ *and* $t > 1$ *we have* $f(x,t) = (1-x)^t + xt > 1$.

Proof. On differentiating f with respect to x we obtain

$$\frac{\mathrm{d}f}{\mathrm{d}x} = t - t(1-x)^{t-1} > 0,$$

since, when $t > 1$, we have $0 \leqslant (1-x)^{t-1} < 1$. The result now follows, since $f(0,t) = 1$. □

In order to analyse Eq. (14), we define the function $h(x,y,t)$ by

$$h(x,y,t) = \frac{1-x^t}{1-y^t},$$

where $0 < x, y < 1$ and $t > 0$. We also define the function $h_1(\varepsilon, y, t)$ by

$$h_1(\varepsilon,y,t) = h(1-\varepsilon y, 1-y, t) = \frac{1-(1-\varepsilon y)^t}{1-(1-y)^t}.$$

Lemma A.2. *For all* $0 < \varepsilon, y < 1$ *and* $t > 1$, $h_1(\varepsilon, y, t)$ *is strictly increasing in both* y *and* ε.

Proof. It is evident that $h_1(\varepsilon,y,t)$ is strictly increasing in ε, so it is sufficient to show that

$$\frac{\partial h_1}{\partial y} > 0.$$

On partially differentiating $\ln h_1$ with respect to y we obtain

$$\frac{1}{h_1}\frac{\partial h_1}{\partial y} = \frac{\varepsilon t(1-\varepsilon y)^{t-1}}{1-(1-\varepsilon y)^t} - \frac{t(1-y)^{t-1}}{1-(1-y)^t} = \frac{1}{h_2(\varepsilon,y,t)} - \frac{1}{h_2(1,y,t)},$$

where

$$h_2(\varepsilon,y,t) = \frac{1-(1-\varepsilon y)^t}{\varepsilon t(1-\varepsilon y)^{t-1}} = \frac{(1-\varepsilon y)^{1-t} - 1 + \varepsilon y}{\varepsilon t}.$$

It now suffices to show that $h_2(\varepsilon,y,t)$ is strictly increasing in ε. On partially differentiating h_2 with respect to ε we obtain

$$\frac{\partial h_2}{\partial \varepsilon} = \frac{(1-\varepsilon y)^t + \varepsilon y t - 1}{\varepsilon^2 t(1-\varepsilon y)^t}.$$

This is strictly positive by Lemma A.1, since the denominator is clearly positive. The result now follows. □

Corollary A.3. *Let* $0 < x_1, x_2, y_1, y_2 < 1$ *satisfy the following inequalities*:

$$x_2 < y_2, \qquad y_1 < y_2, \qquad \frac{x_1}{y_1} < \frac{x_2}{y_2}.$$

Then, for all $t \geqslant 1$,

$$\frac{1-(1-x_1)^t}{1-(1-y_1)^t} < \frac{1-(1-x_2)^t}{1-(1-y_2)^t}.$$

Proof. From the given inequalities it easily follows that $x_1 < y_1$ and $x_1 < x_2$. We assume that $t > 1$ since the result is trivial when $t = 1$. Now let $\varepsilon_1 = x_1/y_1$ and $\varepsilon_2 = x_2/y_2$, so $0 < \varepsilon_1 < \varepsilon_2 < 1$. Then, on using Lemma A.2, we have

$$\frac{1-(1-\varepsilon_1 y_1)^t}{1-(1-y_1)^t} < \frac{1-(1-\varepsilon_2 y_1)^t}{1-(1-y_1)^t} < \frac{1-(1-\varepsilon_2 y_2)^t}{1-(1-y_2)^t},$$

concluding the proof. □

The following corollary is immediate from the previous one.

Corollary A.4. *With the same conditions as in Corollary* A.3, *for all* $m > 0$ *and* $q \geqslant 1$,

$$\frac{1-(1-x_1^m)^q}{1-(1-y_1^m)^q} < \frac{1-(1-x_2^m)^q}{1-(1-y_2^m)^q}.$$

Corollary A.5. *If* $0 < x < y < 1,\ m_1 > m_2 > 0$ *and* $q \geqslant 1$ *then*

$$\frac{1-(1-x^{m_1})^q}{1-(1-y^{m_1})^q} < \frac{1-(1-x^{m_2})^q}{1-(1-y^{m_2})^q}.$$

Proof. Let $t = q$, $x_1 = x^{m_1}$, $x_2 = x^{m_2}$, $y_1 = y^{m_1}$ and $y_2 = y^{m_2}$. Then, since $0 < x < y < 1$ and $m_1 > m_2 > 0$, it follows that $x_2 < y_2$ and $y_1 < y_2$. Furthermore,

$$\frac{x_1}{y_1} = \left(\frac{x}{y}\right)^{m_1} < \left(\frac{x}{y}\right)^{m_2} = \frac{x_2}{y_2}.$$

The result now follows immediately by Corollary A.3. □

We define a function $\mathcal{J}$, which will be useful in establishing the monotonicity properties of the function $h(x, y, t)$:

$$\mathcal{J}(\theta) = \frac{1-\theta^u}{1-\theta^p},$$

where $0 < \theta < 1$ and $0 < p < u$.

Lemma A.6. *The function* $\mathcal{J}$ *is strictly increasing in* θ *for* $0 < \theta < 1$ *and* $0 < p < u$.

Proof. We show that the derivative of $\mathcal{J}$ with respect to θ is positive. Now

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\theta} = \frac{-u\theta^{u-1}(1-\theta^p) + p\theta^{p-1}(1-\theta^u)}{(1-\theta^p)^2}. \tag{A.1}$$

Since the denominator of (A.1) is positive, it remains to be shown that

$$-u\theta^{u-1} + u\theta^{u+p-1} + p\theta^{p-1} - p\theta^{u+p-1} > 0.$$

On dividing by $up\theta^{u+p-1}$, this is equivalent to the inequality

$$\frac{1}{u}\left(\frac{1}{\theta^u} - 1\right) > \frac{1}{p}\left(\frac{1}{\theta^p} - 1\right).$$

Now let $\beta = 1/\theta$; it is thus sufficient to show that the function $(\beta^z - 1)/z$ is strictly increasing for $z > 0$, i.e., that

$$\frac{\mathrm{d}}{\mathrm{d}z}\left(\frac{1}{z}\left(\beta^z - 1\right)\right) = -\frac{1}{z^2}\left(\beta^z - 1\right) + \frac{1}{z}\beta^z \ln\beta > 0.$$

Multiplying this by $z^2\beta^{-z}$ yields

$$\beta^{-z} > 1 - z\ln\beta. \tag{A.2}$$

Since it is well known that $\mathrm{e}^w > 1 + w$ for all $w \neq 0$, inequality (A.2) follows on setting $w = -z\ln\beta$. □

Corollary A.7. *Let $t > 0$. Then,*

(i) *if $0 < x < y < 1$, the function $h(x, y, t)$ is strictly decreasing in t, and*
(ii) *if $0 < y < x < 1$, the function $h(x, y, t)$ is strictly increasing in t.*

Proof. (i) Let $t_2 > t_1 > 0$. Then, by Lemma A.6,

$$\frac{1 - x^{t_2}}{1 - x^{t_1}} < \frac{1 - y^{t_2}}{1 - y^{t_1}},$$

and thus

$$\frac{1 - x^{t_1}}{1 - y^{t_1}} > \frac{1 - x^{t_2}}{1 - y^{t_2}},$$

yielding the result.

(ii) If $y < x$ then, by part (i), $h(y, x, t)$ is strictly decreasing in t, so its reciprocal $h(x, y, t)$ is strictly increasing in t. □

References

[1] I. Althöfer, I. Leader, Correlation of boolean functions and pathology in recursion trees, SIAM J. Discrete Math. 8 (1995) 526–535.

[2] E.B. Baum, W.D. Smith, A Bayesian approach to relevance in game playing, Artificial Intelligence 97 (1997) 195–242.

[3] D. Beal, M.C. Smith, Random evaluation in Chess, ICCA J. 17 (1994) 3–9.

[4] P.-C. Chi, D.S. Nau, Comparison of the minimax product back-up rules in a variety of games, in: L. Kanal, V. Kumar (Eds.), Search in Artificial Intelligence, Springer, New York, 1988, pp. 450–471.
[5] A.D. de Groot, Thought and Choice in Chess, 2nd edn., Mouton Publishers, The Hague, 1978.
[6] C. Ebeling, All the Right Moves, MIT Press, Cambridge, MA, 1987.
[7] W. Feller, An Introduction to Probability Theory and its Applications, Wiley, New York, 1968.
[8] R.K. Guy, What is a game?, in: R.K. Guy (Ed.), Combinatorial Games, Proceedings of Symposia in Applied Mathematics, Vol. 43, American Mathematical Society, Providence, RI, 1991, pp. 1–21.
[9] D. Hartmann, How to extract relevant knowledge from grandmaster games, Part 2: The notion of mobility, and the work of De Groot and Slater, ICCA J. 10 (1987) 78–90.
[10] D. Hartmann, P. Kouwenhoven, Sundry computer chess topics, in: D.F. Beal (Ed.), Advances in Computer Chess 6, Ellis Horwood, Chichester, 1991, pp. 61–72.
[11] H. Kaindl, Tree searching algorithms, in: T.A. Marsland, J. Schaeffer (Eds.), Computers, Chess and Cognition, Springer, New York, 1990, pp. 133–158.
[12] H. Kaindl, A. Scheucher, Reasons for the effects of bounded look-ahead search, IEEE Trans. Systems Man Cybernet. 22 (1992) 992–1007.
[13] D.E. Knuth, The Art of Computer Programming, Vol. I: Fundamental Algorithms, 2nd edn., Addison-Wesley, Reading, MA, 1973.
[14] K.-F. Lee, S. Mahajan, The development of a world class Othello program, Artificial Intelligence 43 (1990) 21–36.
[15] M. Levene, T.I. Fenner, A partial analysis of minimaxing game trees with random leaf nodes, ICCA J. 18 (1995) 20–33.
[16] D. Levy, M. Newborn, How Computers Play Chess, W.H. Freeman, New York, 1991.
[17] G.P. Michon, Recursive random games: A probabilistic model for perfect information games, Ph.D. Thesis, Computer Science Department, University of California, Los Angeles, CA, 1983.
[18] D.S. Nau, The Last Player Theorem, Artificial Intelligence 18 (1982) 53–65.
[19] D.S. Nau, Decision quality as a function of search depth on game trees, J. ACM 30 (1983) 687–708.
[20] A.J. Palay, Search with Probabilities, Research Notes in Artificial Intelligence, Vol. 3, Pitman Publishing, Marshfield, MA, 1985.
[21] J. Pearl, On the nature of pathology in game searching, Artificial Intelligence 20 (1983) 427–453.
[22] J. Schaeffer, D. Szafron, P. Lu, R. Lake, A re-examination of brute-force search, in: Proc. AAAI Fall Symposium, Games: Planning & Learning, Raleigh, NC, 1993, pp. 51–58.
[23] G. Schrüfer, Presence and absence of pathology in game trees, in: D.F. Beal (Ed.), Advances in Computer Chess 4, Pergamon Press, Oxford, 1986, pp. 101–112.
[24] C.E. Shannon, Programming a computer for playing chess, Philosophical Magazine 41 (1950) 256–275.
[25] E. Slater, Statistics for the chess computer and the factor of mobility, in: Proc. Symposium on Information Theory, London, 1950, pp. 150–152.